

NN	NN	TTTTTTTTTT	000000	DDDDDDDD	AAAAAA	PPPPPPP	CCCCCCC	RRRRRRR	CCCCCCC
NN	NN	TTTTTTTTTT	000000	DDDDDDDD	AAAAAA	PPPPPPP	CCCCCCC	RRRRRRR	CCCCCCC
NN	NN	TT	00	00	DD	AA	PP	RR	RR
NN	NN	TT	00	00	DD	AA	PP	RR	CC
NNNN	NN	TT	00	0C00	DD	AA	PP	RR	CC
NNNN	NN	TT	00	0000	DD	AA	PP	RR	CC
NN	NN	TT	00	00	DD	AA	PP	RR	CC
NN	NN	TT	00	00	DD	AA	PP	RR	CC
NN	NN	TT	00	00	DD	AA	PP	RR	CC
NN	NN	TT	00	00	DD	AA	PP	RR	CC
NN	NNNN	TT	0000	00	DD	AA	PP	RR	CC
NN	NNNN	TT	0000	00	DD	AA	PP	RR	CC
NN	NNNN	TT	0000	00	DD	AA	PP	RR	CC
NN	NNNN	TT	0000	00	DD	AA	PP	RR	CC
NN	NNNN	TT	0000	00	DD	AA	PP	RR	CC
NN	NNNN	TT	0000	00	DD	AA	PP	RR	CC
NN	NNNN	TT	00	00	DD	AA	PP	RR	CC
NN	NNNN	TT	000000	DDDDDDDD	AA	AA	PP	RR	RR
NN	NNNN	TT	000000	DDDDDDDD	AA	AA	PP	RR	RR

LL	IIIIII	SSSSSS
LL	IIIIII	SSSSSS
LL	II	SS
LLLLLLLL	IIIIII	SSSSSS
LLLLLLLL	IIIIII	SSSSSS

(2) 67
(3) 98
(4) 128
(5) 172

DECLARATIONS
NT\$CRC_TABLE - CRC POLYNOMIAL TABLE
NT\$CRC_INIT - INITIALIZE CRC COMPUTATION
NT\$CRC_LOGERR - LOG DAP CRC ERROR

0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :--

Add NT\$CRC_LOGGER routine to log DAP CRC errors to the
DECnet Event Logger (EVL).

V02-002 JAK0058 J A Krycka 01-JUN-1981
This module was created from code previously residing in
NTOCREATE.

```
0000 67 .SBTTL DECLARATIONS
0000 68
0000 69 :
0000 70 : Include Files:
0000 71 :
0000 72
0000 73 $DAPPLGDEF : Define DAP prologue symbols
0000 74 $DAPCNFDEF : Define DAP Configuration message
0000 75 $DAPCRCDEF : Define DAP CRC checksum symbols
0000 76 $EVCFDEF : Define event class symbols
0000 77 $IFBDEF : Define IFAB symbols
0000 78 $IODEF : Define IOS symbols
0000 79 $NFBDEF : Define network function definition symbols
0000 80 $NMADEF : Define network management symbols
0000 81 $NWADEF : Define network work area symbols
0000 82 $RAWDEF : Define raw event record format
0000 83
0000 84 :
0000 85 : Macros:
0000 86 :
0000 87 : None
0000 88 :
0000 89 : Equated Symbols:
0000 90 :
0000 91 : None
0000 92 :
0000 93 : Own Storage:
0000 94 :
0000 95 : None
0000 96 :
```

```
0000 98 .SBTTL NT$CRC_TABLE - CRC POLYNOMIAL TABLE
0000 99
0000 100 ;++
0000 101 ; This is the CRC table for use in DAP CRC checksum computation.
0000 102 ; The CRC polynomial function (order 16) used is:
0000 103 ;  
          x**16 + x**15 + x**13 + x**7 + x**4 + x**2 + x**1 + 1
0000 104 ;
0000 105 ;--
0000 106
0000 107 .ALIGN LONG
0000 108
0000 109 NT$CRC_TABLE:: : CRC polynomial table
00000000 0000 110 .LONG DAP$K_CRC_TBL0 : Table entry 0
000053E3 0004 111 .LONG DAP$K_CRC_TBL1 : Table entry 1
0000A7C6 0008 112 .LONG DAP$K_CRC_TBL2 : Table entry 2
0000F425 000C 113 .LONG DAP$K_CRC_TBL3 : Table entry 3
00009D87 0010 114 .LONG DAP$K_CRC_TBL4 : Table entry 4
0000CE64 0014 115 .LONG DAP$K_CRC_TBL5 : Table entry 5
00003A41 0018 116 .LONG DAP$K_CRC_TBL6 : Table entry 6
000069A2 001C 117 .LONG DAP$K_CRC_TBL7 : Table entry 7
0000E905 0020 118 .LONG DAP$K_CRC_TBL8 : Table entry 8
0000BAE6 0024 119 .LONG DAP$K_CRC_TBL9 : Table entry 9
00004EC3 0028 120 .LONG DAP$K_CRC_TBLA : Table entry 10
00001D20 002C 121 .LONG DAP$K_CRC_TBLB : Table entry 11
00007482 0030 122 .LONG DAP$K_CRC_TBLC : Table entry 12
00002761 0034 123 .LONG DAP$K_CRC_TBLD : Table entry 13
0000D344 0038 124 .LONG DAP$K_CRC_TBLE : Table entry 14
000080A7 003C 125 .LONG DAP$K_CRC_TBLF : Table entry 15
0040 126
```

```

0040 128      .SBTTL  NT$CRC_INIT - INITIALIZE CRC COMPUTATION
0040 129
0040 130 :++
0040 131 : NT$CRC_INIT - performs DAP CRC computation initialization.
0040 132
0040 133 : Calling Sequence:
0040 134
0040 135 :     BSBW  NT$CRC_INIT
0040 136
0040 137 : Input Parameters:
0040 138
0040 139 :     R7      NWA (=DAP) address
0040 140
0040 141 : Implicit Inputs:
0040 142
0040 143 :     DAP$Q_SYSCAP
0040 144
0040 145 : Output Parameters:
0040 146
0040 147 :     R0      Status code (RMS)
0040 148
0040 149 : Implicit Outputs:
0040 150
0040 151 :     DAP$L_CRC_RSLT
0040 152
0040 153 : Completion Codes:
0040 154
0040 155 :     Standard RMS completion codes
0040 156
0040 157 : Side Effects:
0040 158
0040 159 :     None
0040 160
0040 161 :--
0040 162
0040 163 NT$CRC_INIT:: : Entry point
50  D4 0040 164 CLRL   R0 : Initially set status to failure
15  E1 0042 165 BBC    #DAP$V_DAPCRC,- : Branch if partner does not support
0B 28 A7 0044 166 DAP$Q_SYSCAP(R7),10$ : file level CRC checksum
0000FFFF 8F  D0 0047 167 MOVL   #DAP$R_CRC_INIT,- : Use initial CRC value as first
20 A7 004D 168 DAP$L_CRC_RSLT(R7) : CRC resultant value
004F 004F 169 RMSSUC : Return success
05  0052 170 10$: RSB : Exit with status code in R0

```

0053 172 .SBTTL NT\$CRC_LOGERR - LOG DAP [CRC] ERROR
 0053 173
 0053 174 :++
 0053 175 : NT\$CRC_LOGERR - generates an appropriate message to the DECnet Event Logger
 0053 176 : TEVL) to log the occurrence of a DAP level CRC checksum error.
 0053 177
 0053 178 Calling Sequence:
 0053 179
 0053 180 BSBW NT\$CRC_LOGERR
 0053 181
 0053 182 Input Parameters:
 0053 183
 0053 184 R7 NWA (=DAP) address
 0053 185 R8 RAB address
 0053 186 R9 IRAB address
 0053 187 R10 IFAB address
 0053 188 R11 Impure area address
 0053 189
 0053 190 Implicit Inputs:
 0053 191
 0053 192 NWAST_NODEBUF
 0053 193 NWASB_NODBUFSIZ
 0053 194
 0053 195 Output Parameters:
 0053 196
 0053 197 None
 0053 198
 0053 199 Implicit Outputs:
 0053 200
 0053 201 None
 0053 202
 0053 203 Completion Codes:
 0053 204
 0053 205 None
 0053 206
 0053 207 Side Effects:
 0053 208 An attempt is made to log a DAP CRC error to the DECnet Event Logger.
 0053 209 Return status of the request is neither checked nor returned.
 0053 210
 0053 211
 0053 212 :--
 0053 213
 0053 214 NT\$CRC_LOGERR:: : Entry point
 3F 83 0053 215 PUSHR #^M<R0,R1,R2,R3,R4,R5> : Save registers
 0055 216
 0055 217 : Get the name of the partner node
 0055 218
 52 0168 C7 90 0055 219 MOVB NWASB_NODBUFSIZ(R7),R2 : Get size of nodename
 53 0169 C7 9E 005A 220 MOVAB NWAST_NODEBUF(R7),R3 : Get address of node spec list
 005F 221
 005F 222
 005F 223 : Build the NFB descriptor, the NFB, the event buffer descriptor and
 005F 224 : the event buffer itself.
 005F 225
 005F 226
 55 0120 C7 9E 005F 227 MOVAB NWAST TEMP(R7),R5 : Get address to build event buffer
 85 05 9A 0064 228 MOVZBL #5,(R5)+ : NFB desc count field

85	04	A5	DE	0067	229	MOVAL	4(R5), (R5)+	: NFB desc addr field
85	1C	90	006B	230	MOVB	#NFB\$C_LOGEVENT, (R5)+	: NFB	
85	85	D4	006E	231	CLRL	(R5)+		
85	52	22	C1	0070	ADDL3	#RAWST DATA+4, R2, (R5)+	: Event buffer desc count	
85	04	A5	DE	0074	MOVAL	4(R5), (R5)+	: Event buffer desc address	
85	F8	A5	80	0078	MOVW	-8(R5), (R5)+	: Put event count in event buffer	
				007C	SGETTIM_S -		: Get the system time	
				007C	TIMADR=(R5)		: Put time in the event buffer	
85	55	08	C0	0085	ADDL2	#8, R5	: Bump the event buffer pointer	
85	2001	8F	80	0088	MOVW	#EV\$C_VMS_DPC, (R5)+	: Put in the event code	
85	FF	8F	90	008D	MOVB	#EV\$C_SRC_NON, (R5)+	: Put in the source type	
85	55	11	C0	0091	ADDL2	#17, R5	: Skip the source ID field	
85	85	00	B0	0094	MOVW	#EV\$C_VMS_PNOD, (R5)+	: Remote node name	
85	40	8F	90	0097	MOVB	#NMASM_PTY_ASC, (R5)+	: Data type	
85	85	52	90	0098	MOVB	R2, (R5)+	: Put in the nodename count	
65	63	52	28	009E	MOVC3	R2, (R3), (R5)	: Put in the nodename	
55	0120	C7	9E	00A2	MOVAB	NWAST TEMP(R7), R5	: Get address of TEMP	
54	0D	A5	9E	00A7	MOVAB	13(R5), R4		
				00AB	SQIO_S -			
				00AB	CHAN=IFBSW CHNL(R10), -		: Use already assigned channel	
				00AB	FUNC=#IOS\$_ACPCONTROL, -		: Function code	
				00AB	P1=(R5), -		: Adr of descriptor of NFB	
				00AB	P2=R4		: Adr of descriptor of event buffer	
				00C7	252			
				00C7	253			
				00C7	254		: Do not bother to check the return status since we wish to preserve the	
				00C7	255		: original DAP CRC error status in R0 (and not the success or failure of	
				00C7	256		: our attempt to log it).	
				00C7	257			
				00C7	258			
3F	BA	00C7	259	30\$. POPR	#^M<R0,R1,R2,R3,R4,R5>	: Restore registers		
05		00C9	260	RSB		: Exit		
		00CA	261					
		00CA	262	.END		: End of module		

\$\$PSECT EP	= 00000000	NT\$CRC_INIT	00000040	RG	01
\$\$RMSTEST	= 0000001A	NT\$CRC_LOGERR	00000053	RG	01
\$\$RMS_PBUGCHK	= 00000010	NT\$CRC_TABLE	00000000	RG	01
\$\$RMS_TBUGCHK	= 00000008	NWA\$B_ALLXABCNT	0000011C		
\$\$RMS_UMODE	= 00000004	NWA\$B_DAP_RAC	000000C9		
SST1	= 00000001	NWA\$B_FILESYS	000000C5		
DAP\$B_DCODE_FID	= 00000019	NWA\$B_KEYXABCNT	0000011D		
DAP\$B_DCODE_MAC	= 0000001B	NWA\$B_NETSTRSIZ	0000016F		
DAP\$B_DCODE_MSG	= 0000001A	NWA\$B_NODBUFSIZ	00000168		
DAP\$B_DECVER	= 00000047	NWA\$B_ORG	000000C6		
DAP\$B_ECONUM	= 00000045	NWA\$B_OSTYPE	000000C4		
DAP\$B_FILESYS	= 00000043	NWA\$B_RFM	000000C7		
DAP\$B_OSTYPE	= 00000042	NWA\$B_RMS_RAC	000000C8		
DAP\$B_USRNUM	= 00000046	NWA\$C_BLN	00000800		
DAP\$B_USRVER	= 00000048	NWA\$K_BLN	00000800		
DAP\$B_VERNUM	= 00000044	NWA\$L_ALLXABADR	00000100		
DAP\$B_X_FIELD	= 00000024	NWA\$L_DATXABADR	00000104		
DAP\$C_BCN	= 000000C0	NWA\$L_DEV	000000C0		
DAP\$K_BLN	= 000000C0	NWA\$L_FHCXABADR	00000108		
DAP\$K_CRC_INIT	= 0000FFFF	NWA\$L_KEYXABADR	0000010C		
DAP\$K_CRC_TBL0	= 00000000	NWA\$L_MSG_MASK	000000D4		
DAP\$K_CRC_TBL1	= 000053E3	NWA\$L_PRDXABADR	00000110		
DAP\$K_CRC_TBL2	= 0000A7C6	NWA\$L_RDTXABADR	00000114		
DAP\$K_CRC_TBL3	= 0000F425	NWA\$L_SAVE_FLGS	00000128		
DAP\$K_CRC_TBL4	= 00009D87	NWA\$L_SUMXABADR	00000118		
DAP\$K_CRC_TBL5	= 0000CE64	NWA\$L_THREAD	000000FC		
DAP\$K_CRC_TBL6	= 00003A41	NWA\$L_XLTATTR	00000238		
DAP\$K_CRC_TBL7	= 000069A2	NWA\$L_XLTBUFLG	0000022C		
DAP\$K_CRC_TBL8	= 0000E905	NWA\$L_XLTCNT	00000228		
DAP\$K_CRC_TBL9	= 0000BAE6	NWA\$L_XLTMAXINDX	00000234		
DAP\$K_CRC_TBLA	= 00004EC3	NWA\$L_XLTSIZ	00000230		
DAP\$K_CRC_TBLB	= 00001D20	NWA\$Q_ACS	00000244		
DAP\$K_CRC_TBLC	= 00007482	NWA\$Q_BIGBUF	00000170		
DAP\$K_CRC_TBLD	= 00002761	NWA\$Q_BLD	000000F0		
DAP\$K_CRC_TBLE	= 0000D344	NWA\$Q_FLG	00000000		
DAP\$K_CRC_TBLF	= 000080A7	NWA\$Q_INODE	0000025C		
DAP\$L_CMWA	= 00000030	NWA\$Q_IOSB	000000D8		
DAP\$L_CRC_RSLT	= 00000020	NWA\$Q_LICDF	00000160		
DAP\$L_DCODE_STS	= 00000018	NWA\$Q_LOGNAME	0000023C		
DAP\$L_MSG_MASK	= 0000C01C	NWA\$Q_NCB	00000264		
DAP\$L_SSPWA	= 00000080	NWA\$Q_RCV	000000E0		
DAP\$L_TEMP	= 00000090	NWA\$Q_SAVE_DESC	00000120		
DAP\$Q_DCODE_FLG	= 00000000	NWA\$Q_XLTBUF1	0000024C		
DAP\$Q_MSG_BUF1	= 00000008	NWA\$Q_XLTBUF2	00000254		
DAP\$Q_MSG_BUF2	= 00000010	NWA\$Q_XMT	000000E8		
DAP\$Q_SYSCAP	= 00000028	NWA\$T_ACBSUF	0000026C		
DAP\$V_DAPCRC	= 00000015	NWA\$T_AUXBUF	000005E0		
DAP\$W_BUFSIZ	= 00000040	NWA\$T_DAP	00000000		
DAP\$W_PARTNER	= 00000006	NWA\$T_INODEBUF	000004AC		
DAP\$W_VERSION	= 00000004	NWA\$T_ITM_ATTR	00000200		
EVCSC_SRC_NON	= 000000FF	NWA\$T_ITM_END	00000224		
EVCSC_VMS_DPC	= 00002001	NWA\$T_ITM_LST	00000200		
EVCSC_VMS_PNOD	= 00000000	NWA\$T_ITM_MAXINDX	00000218		
IFBSW_CHNC	= 00000020	NWA\$T_ITM_STRING	0000020C		
IOS_APPCONTROL	= 00000038	NWA\$T_NCBBUF	0000052C		
NFBSC_LOGEVENT	= 00000001C	NWA\$T_NODEBUF	00000169		
NMASM_PTY_ASC	= 00000040	NWA\$T_RCVBUF	000001A0		

NWAST_SCAN	00000100
NWAST_TEMP	00000120
NWAST_XLTBUF1	000002AC
NWAST_XLTBUF2	000003AC
NWAST_XMTBUF	000003C0
NWASW_BUILD	000000D2
NWASW_DAPBUFSIZ	000000CA
NWASW_DIR OFF	000000CC
NWASW_DISPLAY	000000D0
NWASW_FIL OFF	000000CE
NWASW_JNLXABJOP	0000011E
RAWSB_SRCTYP	0000000C
RAWSC_SIZE	0000001F
RAWSK_SIZE	0000001F
RAWST_DATA	0000001E
RAWST_SRCID	0000000D
RAWST_SYSTIM	00000002
RAWSW_BYTES	00000000
RAWSW_EVTCODE	0000000A
SYSSGETTIM	*****
SYSSQIO	*****

01
01

! Psect synopsis !

PSECT name

Allocation	PSECT No.	Attributes											
00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	I	CL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
000000CA (202.)	01 (1.)	PIC	USR	CON	REL	3	BL	NOSHR	EXE	RD	NOWRT	NOVEC	LONG
00000800 (2048.)	02 (2.)	NOPIC	USR	CON	ABS	L	CL	NOSHR	EXE	RD	WRT	NOVEC	BYTE

! Performance indicators !

Phase

Page faults	CPU Time	Elapsed Time
34	00:00:00.10	00:00:00.95
161	00:00:01.15	00:00:06.08
624	00:00:22.02	00:01:00.88
0	00:00:03.37	00:00:06.64
84	00:00:03.18	00:00:08.96
19	00:00:00.17	00:00:00.23
3	00:00:00.03	00:00:00.16
0	00:00:00.00	00:00:00.00
928	00:00:30.03	00:01:23.91

The working set limit was 1650 pages.

114649 bytes (224 pages) of virtual memory were used to buffer the intermediate code

There were 130 pages of symbol table space allocated to hold 2403 non-local and 2 local symbols.

262 source lines were read in Pass 1, producing 13 object records in Pass 2

202 source lines were read in pass 1, producing 13 object
25 pages of virtual memory were used to define 24 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

-\$255\$DUA28:[SYS.OBJ]LIB.MLB:1
-\$255\$DUA28:[SHRLIB]NMALIBRY.MLB:1
-\$255\$DUA28:[SHRLIB]EVCDEF.MLB:1
-\$255\$DUA28:[RMS.OBJ]RMS.MLB:1
-\$255\$DUA28:[SYSLIB]STARLET.MLB:2
TOTALS (all libraries)

Macros defined

2
0
2
7
9
20

2635 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LI\$S:NTODAPCRC/OBJ=OBJ\$S:NTODAPCRC MSRC\$S:NTODAPCRC/UPDATE=(ENH\$S:NTODAPCRC)+LIB\$S:RMS/LIB+SHRLIB\$S:EVCDEF/LIB+SHRLIB\$S:NMALIBRY

0315 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

NT0ACCESS
LIS

NT0BLDXAB
LIS

NT0CLOSE
LIS

NT0CONN
LIS

NT0CREATE
LIS

NT0DAPIO
LIS

NT0DAPRC
LIS

NT0BLKTO
LIS

NT0BLKTO
LIS